TL

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/821,687	04/09/2004	F. Soner Terek	MSFT-2955/307064.01	1355

41505 7590 06/28/2006

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
ONE LIBERTY PLACE - 46TH FLOOR
PHILADELPHIA, PA 19103

EXAMINER	
LIANG, GWEN	
ART UNIT	PAPER NUMBER
2162	

DATE MAILED: 06/28/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/821,687	**Applicant(s)** TEREK ET AL.
	Examiner GWEN LIANG	**Art Unit** 2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 May 2006*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *38,40-49 and 54-58* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *38,40-49 and 54-58* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to communications through the applicant's amendment, and Request for Continued Examination (RCE) filed on 5/2/2006.

Claim Objections

2. Claims 40-43 and 49, 54 are objected to because of the following informalities:

With regard to independent claim 49, the articles "the a", in line 11, is an improper use of English language and a change to "a" is recommended.

Dependent claim 40 is similarly objected based on the reasons given for claim 49.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 38 and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bennion (U.S. Patent No. 5,634,123).

With respect to claim 38, Bennion teaches a computer readable medium bearing a computer readable representation ... comprising:

at least one binary fragment comprising a binary fragment header and a binary fragment payload (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type", wherein a code point is equivalent to a header and one of the types is data-containing records, in which case the header is a binary fragment header; Figure 2, element 205, wherein the Data portion is equivalent to the payload);

wherein the binary fragment header comprises a type field wherein the type field indicates the fragment is a binary fragment (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type.", wherein the data-containing record type is equivalent to a binary fragment type);

and a length field wherein the length field indicates a length of the binary fragment payload (Figure 2.; col. 1, lines 60-62, "A length field facilitates variable data lengths for data-containing records");

wherein the payload comprises a plurality of primitive data members in storage engine record format; (col. 3, lines 19-24, "Thus, a hierarchical structure emerges. Each individual record contained within COMPANY record 600 may be classified as either a container record or a data-containing record. Container records contain other records, while data-containing records contain data. No record is both a container record and a data-containing record"; col. 4, lines 40-43, "By definition, a data-containing record has a "Length" equal to the actual length of the record itself, since a data-containing record cannot contain another record", wherein a data-containing record is a primitive data member)

at least one additional fragment comprising at least one non-primitive member of the object (See at least col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type"; Figure 3, and col. 6, lines 5-18, wherein "a sample string of data bytes representing a series of hierarchically-organized records" illustrates a fragment comprising non-primitive members of the object).

Bennion does not explicitly teach a payload wherein said plurality of primitive data members are all of the primitive data members of the object. However since each record within the COMPANY object may be classified as either a container record or a data-containing record (i.e. a primitive data member) (col. 3, lines 19-22), it is obvious that said plurality of primitive data members are all of the primitive data members of the object.

It would have been obvious to one having ordinary skill in the art at the time the invention was made to include all of the primitive data members of an object in a binary fragment payload as taught by Bennion to provide a complete COMPANY object record as shown in the example given in col. 3 lines 19-31. One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Claim 44 is rejected for the reasons set forth hereinabove for claim 38 and furthermore Bennion discloses a medium comprising a terminator fragment that marks the end of the object, said terminator fragment comprising a terminator type field indicating the terminator fragment is a terminator fragment (col. 6, lines 40-42).

5. Claims 40-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bennion (U.S. Patent No. 5,634,123), further in view of Krishnaprasad et al, "Krishnaprasad" (U.S. Publication No. 2004/0220946), and further in view of Sarkar (U.S. Patent No. 6,012,067),

Claim 40 is rejected for the reasons set forth hereinabove for claim 38. However Bennion does not explicitly disclose a fragment comprising Large Object (LOB) fragment.

Krishnaprasad discloses a computer readable medium comprising:

at least one Large Object (LOB) fragment comprising a LOB fragment header (page 5, section [0062], "Message 300 includes four fields: a length field 302; a version field 304; a flag field 306; and a payload field 310. The combination of fields 302, 304, 306, 310 composes a serialized image of XML data, according to the illustrated embodiment", wherein the fields except the payload field are considered to be the header portion) and a LOB fragment payload (page 6, section [0066], "Payload field 310 includes serialized XML data for a particular XML construct");

wherein the LOB header comprises a LOB type field, wherein the LOB type field indicates the LOB fragment is a LOB fragment (page 6, section [0066], "If the flag field

306 indicates that the type is LOB, then a locator for the LOB appears in the payload, such as the octal string "0.times.000030303030303- 0 . . . ");

and a LOB length field, wherein the LOB length field indicates the a length of the LOB fragment payload (page 5, section [0063], "Length field 302 includes data that indicates the length of the serialized image. Any method known in the art for indicating length may be used."; Page 6, TABLE 2).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a LOB fragment structure as disclosed by Krishnaprasad into the computer readable representation as disclosed in Bennion so that an XML element may be transferred between processes 132a and 132b, which both have access to LOB 144b, by passing the LOB locator (page 4, section [0047]). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

However the combination of Bennion and Krishnaprasad does not explicitly disclose a value type field that indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location.

Sarkar teaches a value type field, wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location (col. 4, lines 25-33, "Internal LOB columns contain LOB locators that can refer to out-of-line or inline LOB values. Selecting a LOB column value returns the LOB locator and not the entire LOB value. Different operations in the form of packages and functions are performed through these locators. Multiple LOB data type columns can be defined in a table and all possible SQL operations are possible over such tables and attributes. LOB locator can be stored in the table column, either with or without the actual LOB value").

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a value type field as disclosed by Sarkar into the computer readable representation as disclosed in the combination of Bennion and Krishnaprasad so that when a LOB column value is selected, the LOB locator is first returned instead of the entire LOB value (col. 4, lines 27-28). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Claim 41 is rejected for the reasons set forth hereinabove for claim 40 and furthermore Sarkar discloses a medium wherein the LOB fragment payload comprises a LOB (col. 4, lines 25-33).

Claim 42 is rejected for the reasons set forth hereinabove for claim 40 and furthermore Sarkar discloses a medium wherein the LOB fragment payload comprises a pointer to a LOB location (col. 4, lines 25-33).

Claim 43 is rejected for the reasons set forth hereinabove for claim 40 and furthermore Sarkar discloses a medium wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB, a pointer to a LOB location, or a cell reference (col. 4, lines 25-33; col. 5, lines 18-47).

6. Claims 45-48, 55-57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bennion (U.S. Patent No. 5,634,123), and further in view of Stickler (US Patent No. 6,904,454 B2).

Claim 45 is rejected for the reasons set forth hereinabove for claim 38 and furthermore Bennion discloses a medium wherein said at least one additional fragment comprises:

a collection start fragment comprising a collection start header (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type", wherein container records are equivalent to "collection". A code point is equivalent to the start header);

wherein the collection start header comprises a collection start type field, wherein the collection start type field indicates the collection start fragment is a collection start fragment (col. 1, lines 56 - 63, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type. A length field facilitates variable data lengths for data-containing records, as well as implicit definition of a hierarchical structure among records", wherein container records are equivalent to "collection". A code point is equivalent to the start header. "a hierarchical structure among records" indicates the structure of a collection type record; col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)"); and

a plurality of collection element fragments (See at least col. 1, lines 56 – 63; Figure 3, and col. 6, lines 5-18).

However Bennion does not explicitly disclose a bit field, wherein the bit field indicates whether an order exists among a plurality of collection element fragments.

Stickler discloses a bit field, wherein the bit field indicates whether an order exists among a plurality of collection element fragments (col. 1, lines 56-58, "at least one entity includes metadata that identifies a sequential relationship between one or more entities within the scope of said one entity, each of said entities including metadata defining a position within said sequential relationship", wherein the one entity that identifies a sequential relationship (i.e. order of entities) is analogous to the bit field that indicates whether an order exists among collection element fragments as claimed).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a bit field indicating whether an order exists as disclosed by Stickler among a plurality of collection element fragments as disclosed in Bennion to overcome a difficult situation present in known tree-based versioning models namely their inability to explicitly define relationships between different releases (col. 2, lines 1-5) Furthermore, MARS 25 also provides encoding properties defining special qualities relating to the format, structure or general serialization of data streams (col. 10, lines 16-18). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Claim 46 is rejected for the reasons set forth hereinabove for claim 45 and furthermore Bennion discloses a medium comprising:

at least one collection element fragment comprising a collection element header and collection element payload (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type"; col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)", wherein a code point is equivalent to a header and one of the types is container records, in which case the header is a collection element header because container records contain other records, which is equivalent to a "collection" element, and the data itself is the payload);

wherein the collection element header comprises a collection element type field, wherein the collection element type field indicates the collection element fragment is a collection element fragment; (col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)",

and a collection element length field, wherein the collection element length field indicates the a length of the collection element payload (col. 5, lines 23-25, "For container records, the length field 203 specifies the total length of all records that the current record contains (as described above)");

Claim 47 is rejected for the reasons set forth hereinabove for claim 46, and furthermore Bennion discloses a medium wherein the collection element payload comprises a data member in a collection of data members corresponding to said collection start fragment (col. 3, lines 32-43).

Claim 48 is rejected for the reasons set forth hereinabove for claim 46, and furthermore Bennion discloses a medium wherein

the collection element header further comprises a collection element locator field that provides a unique location of a data member in a collection of data members (col. 4, lines 24-31).

Claim 55 is rejected on grounds corresponding to the reasons given above for claims 45 and 46.

Claims 56, 57 are rejected on grounds corresponding to the reasons given above for claims 47 and 48.

7. Claims 49 and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bennion (U.S. Patent No. 5,634,123), in view of Krishnaprasad et al, “Krishnaprasad” (U.S. Publication No. 2004/0220946), in view of Sarkar (U.S. Patent No. 6,012,067), and further in view of Stickler (US Patent No. 6,904,454 B2).

With respect to claim 49, Bennion teaches a computer readable medium bearing a computer readable representation … comprising:

a collection start fragment comprising a collection start header (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type", wherein container records are equivalent to "collection". A code point is equivalent to the start header);

wherein the collection start header comprises a collection start type field, wherein the collection start type field indicates the collection start fragment is a collection start fragment (col. 1, lines 56 - 63, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type. A length field facilitates variable data lengths for data-containing records, as well as implicit definition of a hierarchical structure among records", wherein container records are equivalent to "collection". A code point is equivalent to the start header. "a hierarchical structure among records" indicates the structure of a collection type record; col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)"); and

a plurality of collection element fragments (See at least col. 1, lines 56 – 63; Figure 3, and col. 6, lines 5-18).

However Bennion does not explicitly disclose a fragment comprising Large Object (LOB) fragment and nor does Bennion explicitly disclose a bit field, wherein the bit field indicates whether an order exists among a plurality of collection element fragments.

Krishnaprasad discloses a computer readable medium bearing ... representation ... comprising:

at least one Large Object (LOB) fragment comprising a LOB fragment header (page 5, section [0062], "Message 300 includes four fields: a length field 302; a version field 304; a flag field 306; and a payload field 310. The combination of fields 302, 304, 306, 310 composes a serialized image of XML data, according to the illustrated embodiment", wherein the fields except the payload field are considered to be the header portion) and a LOB fragment payload (page 6, section [0066], "Payload field 310 includes serialized XML data for a particular XML construct");

wherein the LOB header comprises a LOB type field, wherein the LOB type field indicates the LOB fragment is a LOB fragment (page 6, section [0066], "If the flag field 306 indicates that the type is LOB, then a locator for the LOB appears in the payload, such as the octal string "0.times.000030303030303- 0 . . . ");

and a LOB length field, wherein the LOB length field indicates the a length of the LOB fragment payload (page 5, section [0063], "Length field 302 includes data that indicates the length of the serialized image. Any method known in the art for indicating length may be used."; Page 6, TABLE 2).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a LOB fragment structure as disclosed by Krishnaprasad into the computer readable representation as disclosed in Bennion so that an XML element may be transferred between processes 132a and 132b, which both have access to LOB 144b, by passing the LOB locator (page 4, section [0047]). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

However the combination of Bennion and Krishnaprasad does not explicitly disclose a value type field that indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location, nor a bit field, wherein the bit field indicates whether an order exists among a plurality of collection element fragments

Sarkar teaches a value type field, wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location (col. 4, lines 25-33, "Internal LOB columns contain LOB locators that can refer to out-of-line or inline LOB values. Selecting a LOB column value returns the LOB locator and not the entire LOB value. Different operations in the form of packages and functions are performed through these locators. Multiple LOB data type columns can be defined in a table and all possible SQL operations are possible over such tables and attributes. LOB locator can be stored in the table column, either with or without the actual LOB value").

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a value type field as disclosed by Sarkar into the computer readable representation as disclosed in the combination of Bennion and Krishnaprasad so that when a LOB column value is selected, the LOB locator is first returned instead of the entire LOB value (col. 4, lines 27-28). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

However the combination of Bennion, Krishnaprasad and Sarkar does not explicitly disclose a bit field, wherein the bit field indicates whether an order exists among a plurality of collection element fragments.

Stickler discloses a bit field, wherein the bit field indicates whether an order exists among a plurality of collection element fragments (col. 1, lines 56-58, "at least one entity includes metadata that identifies a sequential relationship between one or more entities within the scope of said one entity, each of said entities including metadata defining a position within said sequential relationship", wherein the one entity that identifies a sequential relationship (i.e. order of entities) is analogous to the bit field that indicates whether an order exists among collection element fragments as claimed).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a bit field indicating whether an order exists as disclosed by Stickler among a plurality of collection element fragments as disclosed in Bennion to overcome a difficult situation present in known tree-based versioning models namely their inability to explicitly define relationships between different releases (col. 2, lines 1-5) Furthermore, MARS 25 also provides encoding properties defining special qualities relating to the format, structure or general serialization of data streams (col. 10, lines 16-18). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Claim 54 is rejected for the reasons set forth hereinabove for claim 53 and furthermore Bennion discloses a medium comprising:

a collection element fragment comprising a collection element header and collection element payload (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type"; col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)", wherein a code point is equivalent to a header and one of the types is container records, in which case the header is a collection element header because container records contain other records, which is equivalent to a "collection" element, and the data itself is the payload);

wherein the collection element header comprises a collection element type field, wherein the collection element type field indicates the collection element fragment is a collection element fragment; (col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)",

and a collection element length field, wherein the collection element length field indicates the a length of the collection element payload (col. 5, lines 23-25, "For container records, the length field 203 specifies the total length of all records that the current record contains (as described above)");

8. Claim 58 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bennion (U.S. Patent No. 5,634,123), and further in view of Roy et al., "Roy" hereinafter, (US Patent No. 6,631,130).

Claim 58 is rejected on grounds corresponding to the reasons given above for claim 38.

However Bennion does not explicitly disclose that the type field indicates that the binary fragment is the only fragment of the object.

Roy discloses that the type field indicates that the binary fragment is the only fragment of the object (col.3, lines 52-59, "The remaining sixty-eight bits of the PDU are used for various other addressing information such as indicating whether the PDU contains an ATM cell, a packet, or a control message, whether reassembly of the packet should be aborted, whether the payload is a first fragment, middle fragment or last fragment, how many payload bytes are in the last fragment, the fragment sequence count, and a destination flow identifier", wherein the fragment sequence count is used to show the fragment count).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a type field indicating that the binary fragment is the only fragment of the object as disclosed by Roy into the binary fragment data representation as disclosed in Bennion for part of addressing information (col. 3, lines 52-54). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Response to Arguments

9. Applicant's arguments with respect to all the pending claims have been considered but are moot in view of the new ground(s) of rejection.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to GWEN LIANG whose telephone number is 571-272-4038. The examiner can normally be reached on 9:30 A.M. - 5:30 P.M. Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JOHN BREENE can be reached on 571-272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

20 June 2006
G.L.

John E. Breene
JOHN BREENE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100